

REMARKS

In the Official Action mailed on **22 May 2008**, the Examiner reviewed claims 1-18, and 28-35. Examiner rejected claims 1-7, 10-16, and 38-35 under 35 U.S.C. § 103(a) based on Kwong (U.S. Patent No. 6,289,506, hereinafter “Kwong”), and Nizhegorodov (U.S. Patent No. 7,032,216, hereinafter “Nizhegorodov”). Examiner rejected claims 8 and 17 under 35 U.S.C. §103(a) based on Kwong, Nizhegorodov, and Kilis (U.S. Patent No. 5,491,821, hereinafter “Kilis”). Examiner rejected claims 9, and 18 under 35 U.S.C. § 103(a) based on Kwong, Nizhegorodov, and Evans et al. (U.S. Patent No. 5,805,899, hereinafter "Evans").

Rejections under 35 U.S.C. § 103(a)

Claims 1-7, 10 -16, and 28-35 were rejected under 35 U.S.C. § 103(a) as being anticipated by Kwong, in view of Nizhegorodov. Applicant respectfully disagrees.

Kwong discloses optimizing Java performance by selecting a set of program functions in the Java application for native code compilation. These selected program functions are optimized, and then precompiled into native code. Kwong further discloses fine-tuning this process of selection and pre-compilation by reverting from the precompiled native code, and selecting other program functions in the application for native compiling instead (see Kwong, Abstract).

In the Official Action dated 5/22/2008, Examiner cited Figure 7, 710, 715, and 720, Col. 8: 46-47; and Col. 9: 9-11 of Kwong as disclosing the claim recitation, “generating native code from the combined intermediate representation, wherein the native code generation process optimizes interactions between the application running on the virtual machine and the native code method.” Applicant respectfully points out that in the system of Kwong, the

native code that is selected for de-compilation back into the bytecode **refers only to those native code methods that were selected and precompiled initially from the Java bytecode application**. Thus, decompiling these native code methods back into the Java bytecode, and optimizing this bytecode, as disclosed by Kwong, results in an optimization of **only the application bytecode itself**.

Any contextual information that may be used for the optimization involves only that which is present **within the original application bytecode itself**, and does not include contextual information from any native code methods other than those that were originally generated from the application bytecode. To emphasize, in the system of Kwong, there will be no further improvement in the optimization of the application bytecode because of the native code de-compilation back into the bytecode.

In contrast, embodiments of the present invention:

(1) obtain an intermediate representation (IR) for the application program which performs calls to native code methods and callbacks into the VM (see instant application, Fig. 2, step 208); [**Applicant specifically** points out that the native code methods called **here refer to ANY native code, such as from generic native code libraries** (see instant application, paragraphs [0003]-[0004]), and also points out that the application as referred to here only has calls to any native code methods];

(2) obtain an intermediate representation for **any selected native code methods** by decompiling the native code into the IR (see instant application Fig. 2, step 206);

(3) integrate both IRs, where this integration results in a single, combined IR (see instant application, Fig. 2, step 210, and par. [0030]); and

(4) perform optimization on the combined IR while generating native code from the combined IR (see instant application, Fig. 2, step 210, and paragraphs [0028]-[0032]).

Applicant respectfully emphasizes that embodiments of the present invention **combine the equivalent elements into a single intermediate representation and optimize the combined single intermediate representation**. This is beneficial because embodiments of the present invention generate a combined IR that provides additional information to the optimization process that would not be available in the Kwong system.

Consider, for example, a case where a variable, *var*, is set within a native code method from a generic native code library, and execution of a program segment in the application code depends upon the value of this variable, *var*. In embodiments of the present invention, it is possible to **optimize the application code based on the value of *var* in the native code**, since they are both combined into a single IR and available as one entity to the optimization process. This is not possible in the system of Kwong involving calls to any native code method from a Java application.

There is nothing in Kwong, either explicit or implicit, that discloses generating intermediate representations for both the application program as well as any native code methods, integrating both the intermediate representations, and performing an optimization on this integrated intermediate representation. Hence, it is not possible to use the system of Kwong to improve the optimization process by using additional information from the combined IR using both the application program IR as well as the IR for any native code.

Accordingly, Applicant has amended claims 1, 10, 28 and 32 to clarify that an integrated intermediate representation is obtained from the IR for any selected native code method that is called by the application and the IR for the application program, and this integrated IR is optimized. The amendment further clarifies that optimizing involves optimizing calls to the native code method by the application within the integrated intermediate representation, and that the optimizing uses contextual information within the integrated intermediate

representation that is generated from the native code method as well as the application.

In light of these amendments, claims 3 and 12 have been canceled without prejudice. Support for these amendments may be found in Fig. 2, and paragraphs [0003]-[0004], and [0028]-[0032]. No new matter has been added.

Hence, Applicant respectfully submits that independent claims 1, 10, 28, and 32 are in condition for allowance. Applicant also submits that claims 2-9, which depend upon claim 1, claims 11-18, which depend on claim 10, claims 29-31, which depend on claim 28, and claims 33-35, which depend on claim 32, are for the same reasons in condition for allowance and for reasons of the unique combinations recited in such claims.

CONCLUSION

It is submitted that the present application is presently in form for allowance. Such action is respectfully requested.

Respectfully submitted,

By /Anthony Jones/
Anthony Jones
Registration No. 59,521

Date: 22 August 2008

Anthony Jones
Park, Vaughan & Fleming LLP
2820 Fifth Street
Davis, CA 95618-7759
Tel: (530) 759-1666
Fax: (530) 759-1665
Email: tony@parklegal.com